4-1-1991

Computing at Lehigh

Lehigh University

LEHIGH 125 YEARS

LUCC

FOUNDED BY ASA PACKER 1865 · LEHIGH UNIVERSITY · HOMO MINISTER ET INTERPRES NATURAE

Newsletter of the
Lehigh University Computing Center

Volume XIX, Number 4

Contents

From the Director

William R. Harris (WRH0@NS.CC.Lehigh.EDU)

In the last issue of *Computing at Lehigh*, I discussed the work being done to prepare for the replacement of the CYBER and the VAX computers. I mentioned a survey that we planned to send out to users of the systems. That survey has been distributed, and for those of you who would like to express your needs for high-end computing and have not received a copy of the survey, it may be filled out on the Network Server; just enter **NEEDS** at the main LUNA menu. We are attempting to ascertain the present needs and also pent-up computing needs due to the overloaded situation existing on both systems.

We have been collecting benchmark programs from some users of the systems. In addition, we have obtained the SPEC benchmarks and are preparing to run them on the present mainframes. This will allow us to make a rough estimate of the present speed of the mainframes as compared to some of the RISC computers currently on the market. Also, we have subscribed to the AIM benchmark reports which provide an extensive comparison of the processing speed and throughput of most of the popular RISC machines. We will make the reports available in the Computing Center Library for use by those evaluating workstations for purchase.

The Extended Professional Services program has been in existence for about six months and we are presently providing programming support for two groups and possibly providing programming support to a third in the near future. There is a charge involved for the service, but it is strictly on a cost recovery basis. If you are in need of programming assistance for a funded project, we may be able to help at a very reasonable cost.

Progress continues in the high-speed networking area. Telecommunications has recently run optical fiber to four additional buildings: Drown Hall, the Alumni Memorial Building, the Physics building, and Linderman Library. We are installing a new *cisco* router in Packard Lab which we expect to perform better than the present equipment, provide capacity for the new buildings mentioned above, and provide future expansion capacity. These buildings should all be connected to the high-speed network by this Summer although individual offices and rooms in the buildings will not be connected until later.

LUCC is also assisting the College of Business and Economics, the English Writing program, and the Modern Foreign Languages department in the design and installation of new computer labs in Rauch, Drown, and Maginnes. In addition, LUCC will be upgrading the computers in the Central Site computer lab., EWFM 292.

It's going to be another very busy Summer.♦

Lehigh University Computing Center Hardware

CDC CYBER 180 Model 850 (32 MBytes Memory, NOS/VE V1.5.3)
IBM 4381 Model 11 (16 MBytes Memory, VSE/SP V2.1.5)—Administrative
IBM 4381 Model 14 (24 MBytes Memory, VM/SP HPO V1.5.0, MUSIC/SP V2.2)—Network Server
VAX 8530 (32 MBytes Memory, VMS V5.3)

Computing at Lehigh est. 1986

Lehigh University Computing Center Newsletter

Editor ..... Blair R. Bernhardt

194 E.W. Fairchild-Martindale Library and Computing Center #8b
Lehigh University
Bethlehem, PA 18015
(215) 758-3990

General Information

Computing at Lehigh is a report on computing, published four times a year by the Lehigh University Computing Center. Article contributions are primarily by Computing Center staff, although users are also encouraged to contribute. Instructions for submitting articles can be found at the end of this newsletter.

Subscriptions to *Computing at Lehigh* are free of charge to those in the Lehigh University Computing Center user community and to other interested parties. Those who wish to subscribe to *Computing at Lehigh* or make changes regarding their subscription should return the mailing list form included at the end of this newsletter.

Distribution of *Computing at Lehigh* is through Lehigh University campus mail for campus subscribers and through U. S. Postal Service First Class Mail for non-campus subscribers.

Computing at Lehigh is formatted with XEROX's Ventura Publisher™ and printed on a PostScript™ printer.

Public Site Hours (Academic Schedule)

	Room Hours	Student Consulting Hours
Central Site Users' Area, 180 Fairchild-Martindale		
Sun	12:00 noon – 12:00 midn	12:00 noon – 12:00 midn
Mon–Thu	6:30 am – 12:00 midn	8:00 am – 12:00 midn
Fri	6:30 am – 10:00 pm	8:00 am – 5:00 pm
Sat	9:00 am – 8:00 pm	10:00 am – 8:00 pm
Central Site Microlab, 292 Fairchild-Martindale		
Sun	12:00 noon – 12:00 midn	no consulting
Mon–Thu	6:30 am – 12:00 midn	no consulting
Fri	6:30 am – 10:00 pm	no consulting
Sat	9:00 am – 8:00 pm	no consulting
Fritz Lab Annex, Room A3		
Mon–Fri	8:00 am – 10:00 pm	no consulting
Grace, Room 28		
Sun	24 hours	2:00 pm – 12:00 midn
Mon–Thu	24 hours	1:00 pm – 12:00 midn
Fri–Sat	24 hours	1:00 pm – 5:00 pm
Libraries: Fairchild-Martindale, Linderman, & Media Center		
Sun	12:00 noon – 12:00 midn	no consulting
Mon–Sat	8:00 am – 12:00 midn	no consulting
Maginnes, Room 491		
Mon–Fri	8:00 am – 10:00 pm	no consulting
Sat	9:00 am – 1:00 pm	no consulting
Mountaintop Campus, B103 Building A		
Mon–Thu	6:30 am – 10:30 pm	no consulting
Fri	6:30 am – 5:30 pm	no consulting
Mountaintop Campus, D109, D117 Building A		
Mon–Thu	6:30 am – 10:30 pm	no consulting
Fri	6:30 am – 5:30 pm	no consulting
Packard, Room 118		
Mon–Fri	8:00 am – 10:00 pm	10:00 am - noon; 2:00 pm -4:00 pm
Sat	8:00 am – 2:00 pm	no consulting
Packard, Room 502		
Mon–Thu	8:00 am – 10:00 pm	10:00 am – 10:00 pm
Fri	8:00 am – 10:00 pm	10:00 am – 5:00 pm
Sat	8:00 am – 2:00 pm	no consulting
Rauch, Rooms 50, 60		
Mon–Fri	8:00 am – 10:00 pm	no consulting
Whitaker, Room 257		
Mon–Thu	8:00 am – 8:00 pm	no consulting
Fri	8:00 am – 5:00 pm	no consulting

Business Hours

Business Office, 394 Fairchild-Martindale	
Mon–Fri	8:15 am – 12:00 noon 1:00 pm – 4:45 pm
User Services, 185/194/196 Fairchild-Martindale	
Mon–Fri	8:00 am – 12:00 noon 1:00 pm – 5:00 pm
Microcomputer Store, 524 Brodhead Ave.	
Mon–Fri	9:00 am – 5:00 pm
Operations, 171 Fairchild-Martindale	
Mon–Fri	8:00 am – 11:30 am 1:00 pm – 4:30 pm
Operator Support/Machine Room, 179 Fairchild-Martindale	
Sun	2:00 pm – 10:00 pm
Mon–Thu	8:00 am – 12:00 midn
Fri	8:00 am – 10:00 pm
Sat	9:00 am – 5:00 pm

Special Forms Processing Hours

Liquid Ink Plots	
Tue,Fri	8:00 am – until done

Computing Center Directory

Information About Policies and Plans
Director
William R. Harris .... (215) 758-3830
Associate Director
of Computing Consulting Services
Timothy J. Foley 758-3830
Associate Director
of Computing Facilities
Carol D. Lidie 758-3989
Systems Programming Manager
Kevin R. Weiner 758-3991
Microcomputer Store Manager
Robert R. Kendi 758-4606

Information About Bills Received
Administrative Associate
Pamela S. Steigerwalt 758-3825
Accounting Assistant
Annette L. Ruhe 758-3825

Consulting
User Consultants
Blair R. Bernhardt 758-3994
Frederick W. Chapman 758-3218
George J. Grevera 758-5152
Dean E. Nelson 758-4988
Richard A. Silvius 758-3985
Binod K. Taterway 758-3984
Michele A. White 758-3995

Information About Software Availability
Software Librarian - Mainframe
Judith K. Allio 758-3993
Software Librarian - Microcomputer
Doris A. Oravec 758-4592

Systems Status, Technical Information
On-duty Consultant
..... 758-4141

General User Information
Administrative Clerk
Lori F. Hertzog 758-3990
Accounts Coordinator
Ann Marie Matusa 758-3992

Hardware and Computer Site Trouble Reporting - Operations Assistant
Julie Cannici 758-4140

Consulting Policy

Consultants are provided to assist users in the use of Lehigh University's computer resources. Consultants are not authorized to interpret course assignments, write code, or debug program logic.

When in need of a consultation, users are requested to contact the LUCC student consultants (present at several of the public sites and at ext. 84141), who are hired to augment the full-time staff consultants.

Computer	On-Campus Phone (300-19.2K Baud)	Off-Campus Phone (1200/2400 Baud)	Network Node Name	Network
Network Server	(NS) Ext. 46000	974-6000	LEHIGH	BITNET
			NS.CC.Lehigh.EDU	Internet
CYBER 850	(CDC) Ext. 46800	974-6800	CDC1.CC.Lehigh.EDU	Internet
VAX 8530	(VAX) Ext. 46400	974-6400	VAX1.CC.Lehigh.EDU	Internet
Annex Terminal Server	Ext. 44413	974-4413	---	Internet

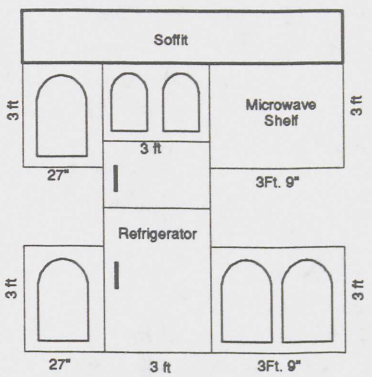
Micro Computing

DrawPerfect - Not just for Fun

Lisa D. Sittler (LL01@NS.CC.LEHIGH.EDU)

Although many people have heard of DrawPerfect by now, most have not used it, thinking that it was just another paint program to try in a spare moment. While DrawPerfect is great for creating drawings, it can do a whole lot more.

DrawPerfect offers many of the usual features for creating illustrations and added features that make it possible to create drawings to scale, as in this kitchen drawing:



Kitchen Cabinet

1 inch equals 3 ft.

DrawPerfect offers a palette of 256 colors, 64 fill patterns, and 16 line/arrow widths and styles, such as solid or dashed. There is a zoom feature to help fine tune your work. Using the *Freehand* option, you can do things such as sign your name or draw a picture. In the standard mode, *Freehand* off, you can easily create circles, rectangles, lines, arcs, and polygons using the appropriate drawing tools. Turning on *Grid display* aids in creating drawings to scale. You can also choose to activate *Grid snap* to increase precision. Naturally, DrawPerfect allows you to create your own grid settings. For example, you could create a grid with points every quarter inch on both the X and Y axes and choose to display every fourth point so that a line drawn from one point to the next on the screen is exactly one inch long. If you choose to turn on *Position display*, DrawPerfect tells you the position of the cursor as you move around the screen and the degrees of rotation when you rotate an object.

DrawPerfect has extensive graph and chart capabilities. You can enter data directly into DrawPerfect or import it from Lotus, Quattro Pro, Excel or delimited ASCII files. You can enter the range coordinates to import or select a named range from a list DrawPerfect displays. DrawPerfect also displays the names of any graphs that are found. You can link your chart to the spreadsheet data so that it will be updated when changes are made to the spreadsheet file. Types of charts include bar, pie, line, scatter, hi-lo, stacked bar, area, and mixed charts. Other options include legends, grid lines, trend lines, 3-D effect, and title and axes labels. Text charts include simple and free-form bullets which are very useful for creating presentation aids. DrawPerfect contains 30 sets of graphics fonts which can be printed on any graphics-capable printer. The font quality depends on the resolution of your printer. Special effects such as boldface, underline, italics, and outline are supported. Text charts can also be enhanced with any of the 500 pre-drawn objects that come standard with DrawPerfect, or any figure of your own design. Flow chart symbols are provided in the standard figure library so this is a natural application for DrawPerfect. Organizational charts are another possible application. It is very simple to re-use objects once you've drawn them using the copy and move options of DrawPerfect.

On-screen *slide show* presentations are another lesser known feature of DrawPerfect. Drawings can be viewed in any sequence you choose, advancing them manually with a mouse or a keystroke, or automatically, by setting a delay period. Various types of screen wipes are offered to clear the screen for the next *slide*. A run-time module comes with DrawPerfect which allows you to create presentations which do not need DrawPerfect to run. Flyers of various service bureaus which create 35mm slides from your DrawPerfect files are enclosed in the DrawPerfect manual.

Finally, as you might guess, DrawPerfect is designed to work hand in hand with WordPerfect. DrawPerfect drawings can be read into WordPerfect directly, and with the Shell program installed (which comes standard with DrawPerfect) users can switch between the programs using a hot key or menu option, without exiting either program. (Be aware that if you don't have a hard drive and some EMS [expanded] memory, the Shell may not work.) Using the two programs in this manner gives you a perfect tool for preparing newsletters, catalogs, documentation, or any type of printed material which mixes graphics with text. If you're a WordPerfect user, you have a headstart in learning DrawPerfect as many of the function keys are the same.

Incidentally, DrawPerfect is still available for faculty and staff from Sandy Edmiston, ext. 84753. Cost for installation on one microcomputer is $23.00; the Reference manual can be purchased for $24.00, and the Figure Library for $14.00. When current stock runs out, expect a slight price increase as WordPerfect has changed their educational program.♦

Borland Upgrades and New Products

Under the Educational Site License Agreement with Borland International, faculty and staff may obtain free-of-charge, but without documentation, Borland software for use on university machines for on-campus use only. Documentation for on-campus use can be ordered through Sandy Edmiston at ext. 84753. Personal copies of Borland products are available through the Microcomputer Store, which includes documentation. The price list can be viewed on the Network Server under INFO topic **PRICES**. The following new or upgraded Borland products are now available:

SIDEKICK V.2.0 - replaces Sidekick V.1.0, but not Sidekick Plus. New features include:

- Mouse support with pull down menus.
- Graphical printout, including PostScript support, for schedules and address books.
- LAN support for shared time planners and address books.
- Reconciliation feature to compare and contrast different sets of appointments.
- 40K RAM-resident space efficiency.

The manual is available, without software, for $10.00. System requirements: DOS 2.0 or higher, 512K RAM, 3 MB of hard disk space.

OBJECTVISION - Borland's first Windows 3.0 application which:

- Allows nontechnical users power to create Windows business applications without traditional programming through use of spreadsheet-like formulas and decision tree logic.
- Provides an administrative computing tool which can be used to create departmental applications such as class registration, benefit administration, and personnel files, etc.

The manual is available, without software, for $25.00. System Requirements: Windows 3.0, 1 MB RAM, 1.5 MB of hard disk space.

OBJECTVISION - RUN TIME - allows an ObjectVision developer to distribute runtime versions of their applications at no additional 'per copy' cost.

- Users of runtime versions of ObjectVision applications can run, but not modify, any application developed with ObjectVision.

The manual is available, without software, for $25.00.

BORLAND C++ - new high-end C and C++ compiler which features:

- Complete C and C++ programming environment for DOS and Windows.
- Virtual Runtime Object-Oriented Memory Manager (VROOMM) which allows DOS applications to break the 640K memory barrier.
- Turbo Drive compiler and integrated development environment which run in protected mode to allow for maximum capacity to compile and debug large programs.
- Turbo Debugger for DOS and Windows.
- Whitewater Resouce Toolkit to visually create resources such as icons, bitmaps, and menu bars for Windows applications.
- Turbo Profiler and Turbo Assembler 2.5 included.

The manual is available, without software, for $50.00. System Requirements: 1 MB of extended memory recommended, 15 MB of hard disk space.

TURBO C++ (2nd Ed.)- repackaged Turbo C++ which offers the same software capabilties as the original Turbo C++.

- Documentation includes original *Programmer's Guide* and combines *Getting Started* and *User's Guide* into one manual. A *Library Reference* manual has been added as part of the on-line help system on the 2nd edition disk set.

These changes allow Borland to lower the manual price from $25.00 to $15.00.

All of these packages are available for downloading from the Network Server under the INFO topic **BORLAND**. Because of the large size of the Borland C++ program, it may be advantageous to obtain it by contacting Sandy Edmiston at ext. 84753.♦

Coming Soon - Upgrades to Borland Products

The following Borland upgrades have been released, but have not yet arrived on campus. Under the site license agreement, faculty and staff may obtain free of charge, but without documentation, Borland software for use on university owned machines for on campus use only. To place an order for Borland manuals, contact Sandy Edmiston at ext. 84753, or user id SJE0, on the Network Server. Personal copies of these products are available through the Microcomputer Store, which includes documentation; type **PRICES** at the main LUNA menu to view the price list.

QUATTRO PRO V. 3.0 - new features include:

- Fully integrated 'what you see is what you get' (WYSIWYG) displays your documents as they'll print.
- WYSIWYG Zoom command reduces or enlarges your screen display.
- Adjustable row heights.
- Snap-to Grid and object alignment tools for easier graphic layout.
- 24 slide-show transition effects for professional presentations.

continued on next page

continued from previous page

- Sound F/X technology adds sounds to your slide shows without special equipment.
- Print-To-Fit automatically prints large spreadsheets on one page.
- Banner printing - prints sideways across continuous paper.
- Easy installation for Windows 3.0.

The manual alone is available for $25.00.

TURBO PASCAL for WINDOWS - includes a fast compiler and the following:

- Windows IDE - edit, compile, and run your programs from within Windows.
- ObjectWindows - built-in support for WINDOWS, dialogs, controls, and more.
- Whitewater Resource Toolkit - create dialogs, menus, icons, bitmaps and other resources.
- Turbo Debugger for Windows - supports WINDOW debugging and reverse execution.
- Turbo Help Hypertext System - complete on-line help system with more than 100 sample programs that you can copy, compile and run immediately.

The manual alone is available for $30.00.

PARADOX ENGINE V.2.xx - Programmers creating custom applications written in C or Pascal, or using DLL libraries under Microsoft WINDOWS, can now integrate the power of Paradox data management into their customized applications. Paradox Engine is the cornerstone of Borland's interoperability. It provides concurrent data sharing across major Borland products, such as Paradox, Quattro Pro, Sidekick, ObjectVision and other third party engine applications

The manual alone is available for $18.00.♦

Mainframe Computing

NOS/VE's Program Development Environment

[*Note: It is anticipated that the CYBER 850 will be taken out of service by September, 1992. Features discussed in this article may not be available after that date.*]

For decades Control Data Corporation's (CDC) CYBER computers have specialized in large-scale, numerically intensive, high-precision scientific computing. Typically, their workloads are as much user-written programs as standard applications and, as a result, an advanced program development environment has evolved.

This article will not be particularly detailed. Its purpose is to highlight all aspects of program development and application tuning on the CYBER 850's operating system, NOS/VE. Detailed help on all these topics can be found in the printed and on-line CDC manuals as well as in the LUCC document entitled the *NOS/VE User's Guide* which is available for purchase at the Lehigh Bookstore and available for reference at most LUCC sites.

The first section of this article explains how to compile and execute simple programs using the **RUN** command and then introduces the full-screen **DEBUG** utility . The second section provides an overview of the Programming Environment, a utility which integrates full-screen editing, context-sensitive help, symbolic debugging, object code performance analysis, and file management. The third section examines the Professional Programming Environment, an environment tailored for complex, multi-person application development that integrates source code and object code library maintenance, and that has an automated build facility.

Anyone already using **RUN** or the Programming Environment, and having experience using the source-level **DEBUG** utility, can skip to the fourth section, Application Optimization and Performance Tuning. Here, tips on how to make code run more efficiently, either by reducing CPU time or page fault activity or both, are provided.

Simple Programs Using the RUN command

Some programs require nothing more than a simple compile and execute. They run and produce a result; they are small and take a short amount of time to complete. Assuming you have one of these simple programs, perhaps coded in FORTRAN, C, Pascal, CYBIL or COBOL, the easiest way to compile and execute it is with the **RUN** command. Suppose a program was created in **EDIT_FILE**, as a file named SIMPLE_PROGRAM. The following command will compile and execute it:

```
RUN  SIMPLE_PROGRAM
```

The first time a program is compiled, **RUN** may not know the language in which it is written. If this is the case, **RUN** will ask for the program's language processor (i.e., the name of the compiler). From then on, **RUN** *knows* the correct compiler to use. In fact, **RUN** also remembers the name of the last program that was run, so just entering

```
RUN
```

will compile and execute SIMPLE_PROGRAM again.

continued on next page

continued from previous page

(As an aside, the **RUN** command can also be used from within **EDIT_FILE**. Just position the cursor on the Home line (by pressing the Home key), and enter **RUN**. **RUN** compiles and executes the current file, and then returns to **EDIT_FILE**.)

If the program completes normally and correctly, everything is finished; otherwise, some debugging needs to be done. This is the time to use the NOS/VE full-screen source-level debug utility **DEBUG**. To use **DEBUG** on SIMPLE_PROGRAM, use the following command:

```
RUN SIMPLE_PROGRAM DEBUG_MODE=ON
```

At this point, **DEBUG** takes control so the program can be easily stepped through, line by line or by setting breakpoints. Variables can be examined or changed dynamically, and watchpoints can be set. Although a full discussion of **DEBUG** is beyond the scope of this article, some of its features include:

- Debugging at the symbolic or the machine level.
- Debugging interactively or at the batch level.
- Debugging interactively in screen mode or line mode.
- Suspending execution of a program when a selected event occurs.
- Displaying the values of selected variables and arrays, and continuing execution.
- Changing the values of program variables while execution is suspended.
- Displaying a subprogram traceback list, beginning with the current subprogram and proceeding back through the sequence of called subprograms until the main program is reached.
- Displaying the current debugging environment.
- Stepping through a program by lines or procedures.
- Creating a file of DEBUG commands that are executed only if an execution error occurs in the program. If an execution error does not occur, the program runs as though DEBUG were not being used.
- Viewing the source code as it executes; an arrow points to the line about to be executed.
- Seeing the output of the program and the messages from DEBUG displayed to the screen.
- Seeing special information about a DEBUG task such as historical or tracing information.
- Viewing each module component of the task.
- Watching program variables as they change value.
- Entering NOS/VE commands.
- Including **DEBUG** commands in SCL procedures.
- Entering commands for processing by another active command processor, such as an editor, to examine the source listing.

Complete information on **DEBUG** can be found in the on-line manual; enter this command:

```
HELP MANUAL=DEBUG
```

or

```
H M=DEBUG
```

The advantage of using **RUN** is its simplicity. The disadvantage is that on-line help functions and program optimization and tuning facilities cannot be readily accessed.

Larger Programs Using the Programming Environment

The Programming Environment integrates many of the NOS/VE full-screen program development utilities: **EDIT_FILE**, **EDIT_CATALOG**, **DEBUG**, **HELP**, and **MEASURE_PROGRAM_EXECUTION**. The Environment also manages multiple programs in any of the supported languages. This means that regardless of whether programs are written in FORTRAN, C, Pascal, or COBOL (or all four languages), the Programming Environment is completely consistent. The command to run the Environment is:

```
ENTER_PROGRAMMING_ENVIRONMENT
```

or

```
ENTPE
```

Once in the Environment, use the Help function key on the various fields; that should be sufficient to get started. Briefly, the Programming Environment allows:

- Getting help on using the Programming Environment.
- Creating and editing programs using the full screen editor, **EDIT_FILE** (**EDIF**).
- Getting information about the language of the program being edited.
- Compiling and running a program.
- Seeing and fixing compilation errors on-line using **EDIF**.
- Seeing and fixing source-related loader errors on-line using **EDIF**.
- Using the full screen interactive debugger **DEBUG**.
- Manipulating compilation options.
- Manipulating files.
- Displaying program performance characteristics.
- Supplying run-time parameters.
- Printing listings of programs.
- Accessing the usage manual of the current programming language.
- Bringing an existing program into the Environment.
- Deleting a program from the Environment.
- Restoring a program that was deleted from the Environment.
- Tailoring the environment defaults for individual needs.

For complete details on the Programming Environment use the Help function key or read the on-line manual:

```
HELP MANUAL=ENVIRONMENT
```

or

```
H M=ENVIRONMENT
```

continued on next page

continued from previous page

Complex Projects Using the Professional Programming Environment

(This section can be safely bypassed by all but advanced users; PPE is not relevant to the casual user.)

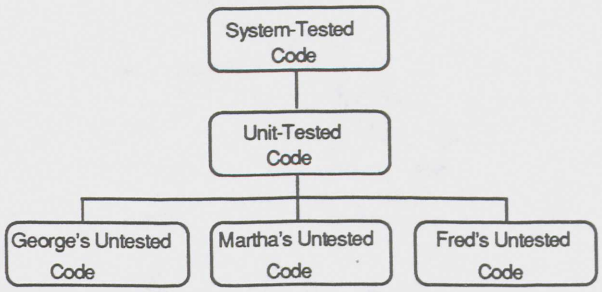
The Professional Programming Environment or PPE performs many of the same functions as the standard Programming Environment, such as transparent access to the editor, on-line help, the debugger, and the file manager. PPE also allows both source and object code to be maintained in a hierarchical structure using **SOURCE_CODE_UTILITY** and **CREATE_OBJECT_LIBRARY**, and allows automated application testing using **BUILD**. PPE is run with the command:

```
ENTER_PPE
```

or

```
ENTPPE
```

In a hierarchical development environment, several versions of the application may exist at one time, each at a different level of the hierarchy. Each level of the hierarchy is a NOS/VE catalog containing code at a certain stage of development, such as untested, unit-tested and system-tested. Each catalog contains all files applicable to its version of code, including a source library, object library, and various listings and data files. The following is an example of a PPE hierarchy where each box represents a NOS/VE catalog:



The lowest level has three catalogs, each containing a developer's untested code.

Suppose Fred, having just joined in the development effort, needs to modify some code. PPE searches up the hierarchy until the specified code is found, makes a copy of it in Fred's source library, and then sets an interlock to prevent others from extracting the same code. Fred is now in complete control of that code section until he has completed his testing and transmitted the new source code back up the hierarchy.

To transmit code means to move it from the level where it currently resides to the next higher level in the hierarchy. In our example, PPE transmits Fred's (now tested) code to the Unit-Tested level of the hierarchy and then deletes the code at Fred's level.

Fred then informs the person in control of the Unit-Tested hierarchy level to rebuild the application for unit-level testing. PPE performs the build and updates the object library in the Unit_Tested level. Indeed, Fred has performed this step numerous times previously during his testing of the application, but the object library updated was his personal copy in his level of the hierarchy.

In this manner, code modifications migrate up the hierarchy until they all reach the System_Tested level and none remain at any of the lower levels.

As usual, full help on PPE may be obtained from the printed and on-line manuals, and by using the Help function key from within PPE:

```
HELP MANUAL=PPE
```

or

```
H M=PPE
```

Application Optimization and Performance Tuning

There are five basic ways to improve an application's efficiency:

- Improve loading.
- Use compiler options.
- Use **AFTERBURN_OBJECT_TEXT**.
- Use **INSTRUMENT_OBJECT_TEXT** and **FORMAT_PROFILE_DATA**.
- Use **MEASURE_PROGRAM_EXECUTION** and read the article "Understanding the Virtual Memory Environment" in the August 1988 issue of *Computing at Lehigh*.

Techniques for improving load efficiency are beyond the scope of this article. In general, this problem will not be experienced unless extremely large applications, like **ANSYS** or **ADINA**, are used. These codes require special binding and pre-linking which are described in the Object Code Management on-line manual:

```
HELP MANUAL=OCM
```

or

```
H M=OCM
```

For typical programs, the first thing to increase efficiency is to use the compiler's optimization parameter. All NOS/VE compilers have an OPTIMIZATION_LEVEL and various debugging parameters. For maximum efficiency, do not turn on any debugging options and set the optimization level to HIGH. One disadvantage of the **RUN** command is that programs cannot be executed at the highest optimization level, so program compilation and execution must be performed manually. To do this (for various compilers):

[**Note:** "/" is the default NOS/VE prompt; ".." continues the command on the next line.]

continued on next page

continued from previous page

```
/FORTRAN INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=HIGH
```

```
/CV2 INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=HIGH
```

```
/PASCAL INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=HIGH
```

```
/COBOL INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=HIGH
```

```
/CYBIL INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=HIGH
```

The second trick to increase efficiency is to try **AFTERBURN_OBJECT_TEXT**, a utility which will inline procedures and perform register optimization. Sometimes, afterburning the object file will make little difference, but it is good practice to at least try it. Assuming SIMPLE_PROGRAM was compiled as described previously, these commands would afterburn and execute the program:

```
/AFTERBURN_OBJECT_TEXT
/$LOCAL.LGO
```

or

```
/AFTOT
/$LOCAL.LGO
```

By default, the afterburner reads and writes the file $LOCAL.LGO, which is the default name of the object file created by the compilers. As with all NOS/VE commands, additional help may be obtained by either of the following methods:

```
HELP AFTERBURN_OBJECT_TEXT   (or H AFTOT)
```

or

```
DISPLAY_COMMAND_INFORMATION AFTERBURN_OBJECT_TEXT
(or DISCI AFTOT)
```

The **INSTRUMENT_OBJECT_TEXT** command modifies an object file so that NOS/VE can count the number of times lines and calls of a program are executed. After execution, use **FORMAT_PROFILE_DATA** to create a report on the data gathered by **INSTRUMENT_OBJECT_TEXT**. Using this data, computing intensive routines can be manually modified. Here is a sample session for SIMPLE_PROGRAM :

```
/FORTRAN INPUT=SIMPLE_PROGRAM ..
   OPTIMIZATION_LEVEL=DEBUG DEBUG_AIDS=ALL
/INSTRUMENT_OBJECT_TEXT
/$LOCAL.LGO
/FORMAT_PROFILE_DATA I=SIMPLE_PROGRAM ..
   LIST=SIMPLE_PROGRAM_PROFILE ..
   LIST_OPTION=(PROGRAM_LISTING LINE_FREQUENCY)
/PRINT_FILE SIMPLE_PROGRAM_PROFILE
```

To gain detailed knowledge of an application's CPU and page fault activity, the **MEASURE_PROGRAM_EXECUTION** (or **MEAPE**) utility is used. **MEAPE** instruments an object file (i.e., adds tracking tools to it), executes it, and then prints a program profile that shows how many times each subroutine is called, how much CPU time it uses, and how many page faults were incurred. Using this data, expensive routines can be hand-optimized. Here is a sample **MEASURE_PROGRAM_EXECUTION** session for the program SIMPLE_PROGRAM:

```
/MEASURE_PROGRAM_EXECUTION
SET_PROGRAM_DESCRIPTION TARGET_TEXT=$LOCAL.LGO
EXECUTE_INSTRUMENTED_TASK
DISPLAY_PROGRAM_PROFILE ..
   OUTPUT=SIMPLE_PROGRAM_PROFILE
QUIT
/PRINT_FILE SIMPLE_PROGRAM_PROFILE
```

MEAPE can also reorganize an object to improve its "locality of reference", a technique that can reduce page faults and improve an application's efficiency. An August 1988 *Computing at Lehigh* article, entitled "Understanding the Virtual Memory Environment", explains the concept of locality of reference and how it can drastically shorten the run time of a program. If excessive page faulting is suspected, please read that article.

As usual, complete help on all these object code optimization commands may be obtained by reading the Object Code Management on-line manual:

```
HELP MANUAL=OCM
```

or

```
H M=OCM
```

Complete help on just command line parameters may be obtained by using **DISPLAY_COMMAND_INFORMATION**.♦

Network Operations

The NREN and Beyond

Building a Nationwide Information Infrastructure

Blair R. Bernhardt (BRB0@NS.CC.LEHIGH.EDU)

The "High-Performance Computing Act of 1991", which is currently before Congress, authorizes the creation of the National Research and Education Network, or NREN. It is the intent of this article to discuss what is meant by the term NREN, and how the NREN is envisioned to impact the national information infrastructure.

Current Nationwide Networks

Independent, but interconnected, nationwide networks are already being run by a number of government agencies including the Defense Advanced Research Projects Agency (DARPA), the Department of Energy (DOE), the National Aeronautics and Space Administration (NASA), and the National Science Foundation (NSF). Some of these networks relate to national security; hence, access to them is restricted. However, the largest and best known of these networks, NSFNET, run by the National Science Foundation, is relatively free of these restrictions. The main purpose behind NSFNET is to provide support for research and education. The main restriction to NSFNET use is that commercial use of the network is prohibited.

So, how does this relate to Lehigh? Lehigh's high-speed backbone network is connected to PREPnet, the Pennsylvania Research and Economic Partnership Network. PREPnet, in turn, is connected to NSFNET through the Pittsburgh Supercomputing Center. PREPnet spans the entire state of Pennsylvania. An ever increasing number of educational institutions are connected to PREPnet as are a number of commercial entities which provide support to research and education.

PREPnet is referred to as a "regional network". There are a large number of similar regional networks throughout the country, many of which are also connected to NSFNET. NSFNET is basically a nationwide *backbone* network which connects other networks. Through NSFNET, anyone at an institution connected to PREPnet can communicate with anyone else connected to any other regional network which is connected to NSFNET. This is not limited to just this country. NSFNET has over 700 international connections to regional networks outside of the United States. All together, there are over 2,300 of these interconnected regional networks which are collectively referred to as the Internet.

The National Research and Education Network

Since NSFNET is already in place, what will the National Research and Education Network be? The NREN can probably best be thought of as NSFNET on a much grander scale. While NSFNET is currently in the final stages of being upgraded from a transmission rate of 1.5 megabits (1.5 million bits per second) to 45 megabits, the NREN is envisioned as having a transmission rate of at least 1 *giga*bit (1 *billion* bits per second). There is some disagreement as to what is meant by the statement that the NREN will be a gigabit network. NSFNET is currently being upgraded to 45 megabits meaning that a total of 45 million bits of information can flow from one network node to the next in any given second, regardless of the number of individual users of the network sending information. When discussing the NREN, directors in a number of federal agencies envision the transmission rate as 1 gigabit *per user*, and not just 1 gigabit for the entire network. In networking discussions, the term *tera*bit (1 *trillion* bits of information per second) is even beginning to be used to describe transmission rates.

What makes discussions of the NREN even more interesting is that the NREN is envisioned as a *prototype* network. It is envisioned that private enterprise will create similar networks to provide similar capabilities to the commercial sector. These private networks are envisioned as going as far as the home. Just as most homes currently have phone and cable television connections, network connections are envisioned as a reality, possibly by the turn of the century. AT&T and MCI both have already invested major amounts to make this a reality. Gateways between the NREN and these private networks will complete the creation of a nationwide information infrastructure.

Will the NREN Become Reality?

Due to the current extent of NSFNET, some people believe that the NREN is already a reality. Actually, NSFNET is only a fraction of the size envisioned for the NREN and transmits data at only a fraction of the speed. However, the "High-Performance Computing Act of 1991", which is currently before Congress (S. 272), calls for the establishment of a *multi-giga*bit National Research and Education Network by 1996.

With all of the constraints currently being placed on the Federal budget, having a bill before Congress may imply that the NREN is on rather unstable ground. However, Eugene Wong, who has the title of Associate Director of the White House Office of Science and Technology Policy, has publicly stated that it is the view of the White House that no Congressional action is needed in order for the NREN to become reality. The current bill is actually authorization legislation,

continued on next page

continued from previous page
which is not needed. The thing which is needed, but which is not a part of the current legislation, is appropriation legislation which actually supplies the capital to fund the NREN. Even without appropriation legislation, in time the NREN will become a reality. Members of the Federal Networking Council, which is made up of directors from all of the agencies charged with the establishment of the NREN, are all committed to making NREN a reality.♦

General Interest

Maple Campus-Wide License

The University has acquired a Maple Campus-Wide License from Waterloo Maple Software. Under the terms of the license, faculty and staff may install all available implementations of Maple (except the Macintosh) on Lehigh-owned computing systems on campus. [Under the sub-license to the Maple Campus-Wide License, full-time faculty, staff, and students may obtain personal copies of microcomputer versions of Maple (Amiga, Atari, and 386-based systems) from the university Microcomputer Store.]

Maple is a symbolic computation package. Symbolic computation packages provide exact solutions to mathematical problems rather than the approximate, numerical solutions produced by more traditional (floating-point) computing techniques. Maple can be successfully applied to a wide variety of problems, ranging from problems in freshman calculus to problems in the exterior calculus of differential forms. Maple can be used interactively, like a desk calculator, or the user can write Maple programs to perform more involved computations.

Some platforms supported under the Maple Campus-Wide License include: 386-based systems (PC or MS-DOS); DEC VAX/VMS; DEC RISC and Ultrix; HP 9000/300, 400; IBM RISC System/6000; Silicon Graphics IRIS; Sun-3; Sun-4 SPARCstation; and Sun 386i. A complete list of all Maple implementations is available from the LUCC mainframe Software Librarian in Rm. 182 of the Fairchild-Martindale Computing Center.

The Computing Center is the distribution center for all available implementations of Maple for use on Lehigh-owned computing systems on campus. Maple for use on 386-based microcomputer systems is available by contacting Doris Oravec at ext. 84592. All other implementations of Maple (mainframes and workstations) are available by contacting Judy Allio at ext. 83993.

Documentation for Maple consists of the *Maple Reference Mnaual, 5th Edition* and the *Maple First Leaves: A Tutorial Introduction to Maple, 3rd Edition*; both manuals are available for purchase at the Microcomputer Store.♦

Consultant's Corner

Statistical Computing at Lehigh

This is a new regular feature to appear in Consultant's Corner. The purpose of the column will be to address general and specific questions about statistical software available on Lehigh computers, and to relate any new developments at Lehigh with respect to statistical computing. Other bits of information related to statistics and statistical computing may also filter in from time to time that strike the authors as interesting and pertinent.

On March 5th, a presentation on "Bootstrap and Resampling Methods" was given by Drs. Martin Richter and John Gatewood of Psychology and Social Relations respectively. These computer intensive methods are becoming widely used due to the emerging availability of fast, cheap processing power. Dr. Gatewood demonstrated a program he had written to perform a resampling analogue of the common t-test. The presentation was attended by faculty, students, and professional staff from mathematics, education, marketing, economics, social relations, and the Computing Center.

Requests for statistical software, including SAS, SPSS, BMDP, and the S programming language, have been made to the Software Subcommittee of the CCAC. The requests were made for both the UNIX-based PL118 SUN workstations as well as DOS-based microcomputers. Because of the number of requests, a subcommittee has been formed to make a recommendation. Anyone with reason to support or oppose these requests, or to make other comments, is encouraged to contact Dean Nelson (ext. 84988 or DEN0 on the Network Server).

There are several discussion groups on BITNET, which may be of interest. These groups have an international readership and offer a useful resource of statistics and statistical computing experience. Access to these groups is provided on the Network Server. At the LUNA menu, enter

continued on next page

continued from previous page

OFFCAMP. Two menu items are available for subscribing either to BITNET discussion groups (Listserv) or USENET discussion groups. The three groups of interest are the Listserv groups STAT-L, SAS-L, and SPSSX-L. After following the instructions to subscribe, enter **LISTSERV** (for Listserv subscriptions) or **NETNEWS** (for USENET subscriptions) to access recent postings. The SAS-L group is particularly active with a large number of experienced and helpful users.

A joint meeting of SAS users from Lehigh and from Dun and Bradstreet, Inc. will be held at Lehigh on July 17. More details will be forthcoming. A presentation describing the use of SAS at Dun and Bradstreet will be given by members of their programming and development group. Anyone with an interest in SAS and its scope of application is invited to attend. Contact Dean Nelson (ext. 84988 or DEN0 on the Network Server) for more information.

On July 1, the default production version of SAS on the VAX 8530 will become version 6.06, replacing the current default version 5.18. Version 5.18 will be retained on the system at least until January 1992, but users are encouraged to begin using version 6.06 as soon as possible. Until July 1, version 6.06 may be invoked by typing **USE SAS606**. Simply entering **USE SAS** invokes version 5.18. On July 1 and thereafter, **USE SAS** will invoke version 6.06 and **USE SAS518** will invoke version 5.18♦

CCAC Highlights

The Computing Center Advisory Committee (CCAC) charter requires that CCAC meeting "highlights" be reported here, and that the full minutes be available on the Network Server. The minutes on the Network Server may be accessed under INFO topic **CCACMIN**.

Computing Center Advisory Committee Minutes: February 7, 1991

Members Present: W. Brichta, B. Fritchman, T. Foley, R. Gruver, B. Hargreaves, W. Harris, F. Kaczmarczyk, E. Kay, R. Kendi, A. King, R. Lawrence, C. Moses, K. Weiner.

Under LUCC News, two new people have been hired. Brad Price has joined the Computer Store, replacing Al Caruso, and John Troiano has joined the Systems Programming group, providing extended technical services to the University. Another User Services Consultant will be hired as a result of eliminating the User Services Manager position.

The Central Site microcomputers in room 292 have not been replaced as had been planned. Gateway 486 machines are being evaluated as possible machines for the room.

Due to the increased load on the system, the Network Server response time has been slower than before. The load on the system is about 30% greater than the same time last year. The slow response time is not due to the changes made in the system at the beginning of the semester. Ways are being explored to improve response time, but options are very limited.

The VAX and CYBER processors are still running close to capacity. There were questions related to the Five-year Plan and how it fits into other plans within the University. Although the plan is still a draft, it is being followed and has been helpful in getting the amortization funds back into the LUCC budget. It is also helpful to departments and colleges as they develop their own plans that need computing resources.

It was reported that the PC Rollover Subcommittee has met electronically and has distributed four Zenith 158's to the Biology department and one for use in Engineering 1. A memo will be sent to faculty in the near future announcing the availability of funds for new machines. There was a motion to divide the money available into fifteen $3,000 grants. The motion was passed.

It was reported that the Computer Intensive Needs subCommittee has not met for quite a while, but a group within the Computing Center has been meeting weekly, planning for the replacement of the CYBER and the VAX. There will be a meeting in the near future. Matt Reilly will be replacing Bill Schiesser, who is on leave from the university. Roy Herrenkohl has agreed to continue on the subcommittee. LUCC is preparing a Request For Proposal (RFP), collecting benchmarks from some of the users and also collecting some standard industry benchmarks. A survey will be going out to users of the machines in the near future. These items will be discussed with the subcommittee at the next meeting. The schedule that will be proposed will result in the removal of the CYBER by September 1992.

Staff Changes

Michele White recently joined User Services as a User Consultant. Michele came to the Computing Center from Quakertown Community Hospital (QCH) where she worked for the past three and a half years, initially as the microsystems coordinator and more recently as the computer services supervisor. Prior to QCH, Michele worked for Tandy Business Products as a training and support specialist and as a systems engineer. Michele has a B.S. in Computer-Based Management Systems from Clarkson University and is expecting to complete work on an MBA at Moravian College this Summer.♦

Computing at Lehigh Contribution Information

Computing at Lehigh encourages contributions for articles and *Consultant's Corner.*

Contributions can either be submitted electronically via the Network Server to user `BRBO`, or be provided on a MS-DOS formatted diskette. Contributions sent via the Network Server must be in ASCII format (i.e., be plain text). Acceptable document formats are:

- ASCII (not word-processed)
- EXP
- Freestyle
- WordStar
- WordPerfect

Printed copy is welcomed, but please also accompany the printed copy with the text in one of the above formats (especially for articles and other long contributions). All mailed contributions (whether on diskette or printed) should be sent to the following address:

Editor, *Computing at Lehigh*
194 Fairchild-Martindale #8b
Computing Center
Lehigh University
Bethlehem, PA 18015

Contributed articles are included in *Computing at Lehigh* at the discretion of the Computing Center. The Computing Center reserves the right to edit all contributions. Article submissions must be received by July 31st for the August issue; September 15th for the October issue; December 15th for the January issue; and, February 10th for the March issue. Be sure to include your name, mailing address, and phone number.

Computing at Lehigh Mailing List

Check one:

☐ **ADD** my name to the mailing list.
☐ **CHANGE** my address on the mailing list. (List both old and new addresses and be sure to include the Zip Codes.)
☐ **DELETE** my name from the mailing list. (Please include the mailing label or complete address.)

Campus
Name: _____
Dept.: _____
Room & Bldg.: _____
_____

Off-Campus
Name: _____
Address: _____
_____
_____ Zip Code: _____

Old Mailing Address (if changing or deleting):

Return to:

Computing at Lehigh Mailing List
194 Fairchild-Martindale #8b
Computing Center
Lehigh University
Bethlehem, PA 18015